Guiding Natural Language Inference through Lexical Inference Types

Anonymous ACL submission

Abstract

Explanatory natural language inference (NLI) aims to provide a mechanism to produce step-wise explanations by linking premises to conclusions. While previous research efforts have concentrated on generating unconstrained entailment trees, they lack the design of mechanisms for inference control, where the linguistic and symbolic properties of explanations can be guided and better localised within the latent space of the neural NLI model. To improve symbolic control and localisation, in this paper, we start by systematically characterising the symbolic mechanisms behind explanatory sentences, defining a set of explanatory inference types. Next, we formalise the encoder-decoder NLI model as a conditional latent variable model where the dynamics of latent variables can be guided and manipulated via inference types. Empirical results show that the proposed approach can improve inference and retrieval and deliver symbolic control.

1 Introduction

Explanation-based Natural Language Inference (NLI) aims to provide a mechanism to produce explanatory (abductive) inference chains which ground claims to their supporting premises (Thayaparan et al., 2020). This inference process allows a systematic way to support the construction of a step-wise explanation hierarchy, via a particular type of textual entailment which encodes claim-premise relationships at different levels of abstraction. This type of inference is particularly relevant in scientific domains, where reasoning needs to be grounded on sets of known facts and plausible and verifiable abstraction steps. One common way of organising these explanatory chains is with the support of entailment trees (Figure 1), a hierarchical structure linking multiple layers of multiple premises to claims. e.g., in the EntailmentBank (Dalvi et al., 2021).

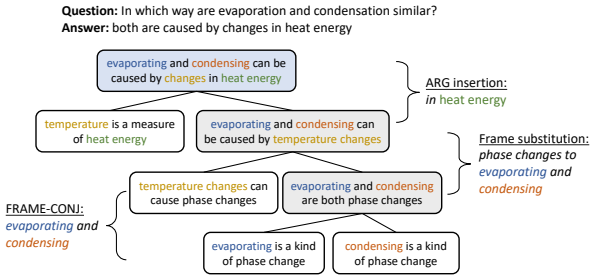


Figure 1: Example of entailment tree. Three steps (1, 2, 3) are present from bottom to top. Each step shows a clear symbolic inference behaviour.

Although Transformer-based language models (Raffel et al., 2020) have been a fundamental component of explanation-based NLI models due to their transferability across different NLI tasks (Thayaparan et al., 2020), the controllability and step-wise interpretability of its internal reasoning have not been fully addressed (Abzianidze, 2023). Additionally, one compounding complexity factor to the problem is the fact that natural language explanatory inference is not fully formalisable within a logical setting, being at the interface between formal and material (content-based) inference (Valentino et al., 2021).

Delivering a controlled explanation-based NLI from a language model thus requires first the characterisation of the formal/symbolic components of explanatory reasoning, from which inference control mechanisms can be derived. For example, within each premise-claim step in Figure 1, there are observable symbolic patterns of inference such as substitution (step 2), conjunction (step 1), and specification (step 3). Those lexical inference patterns (named here *inference types*) reflect a step-wise/localised semantic change that can guide a language model to perform explainable inference in a well-defined, controllable manner, while keeping content-wise flexibility. Specifically, we claim that a representation model aligned to inference

types would allow for improved control and further latent space interpretability. In this work, we focus on the challenge of delivering symbolic control on single-step explanatory NLI (syllogistic-deductive NLI), by targeting the following contributions:

- *A systematic characterisation of explanatory inference types.* To deliver a more controllable and interpretable explanation-based NLI at each entailment step, we start by systematically characterising a set of symbolic patterns and defining associated lexical inference types. The patterns are defined and grounded over a predicate argument-structure representation (Abstract Meaning Representation - AMR symbolic graph) (Banarescu et al., 2013).

- *New NLI mechanisms for integrating inference types within a language model.* Next, to bridge the symbolic and distributional representations, we formalise the encoder-decoder NLI model (T5) as a conditional latent variable model where the inference dynamics over latent variables can be guided and manipulated by our defined inference types. As a complementary, we provide an initialisation of the NLI model with sentence space (named T5bottleneck) to induce the inference information in a low-dimensional latent sentence space.

The empirical analysis indicates that eliciting the inference patterns can assist the training, inference, and in-context learning of the explanatory NLI model. T5bottleneck can improve the explanatory inference retrieval (Valentino et al., 2022a) compared with other latent variable baselines. Moreover, the learned representation can separate different inference types and their symbolic transformation behaviour, indicating the inference rule is encoded in the latent space of the NLI model, implying a quasi-symbolic inference behaviour to the model, where symbolic formal patterns are used in coordination with content-based (distributional inference). The experimental pipelines are released[1].

2 Related Work

In this section, we review the related work around three topics: *neuro-symbolic concepts*, *latent variable model*, *condition(labels)*, to further illustrate our idea and motivation.

Neural-symbolic NLI and concepts. A long-term goal in NLP is to integrate the representation power of neural networks with the interpretability of symbolic or logical systems to build robust NLI models. Current methods often inject symbolic behavior through explicit symbolic representations, including graph (Khashabi et al., 2018; Khot et al., 2017; Jansen et al., 2017; Kalouli et al., 2020; Thayaparan et al., 2021), linear programming (Valentino et al., 2022b; Thayaparan et al., 2024), adopting iterative methods, using sparse encoding mechanisms (Valentino et al., 2020; Lin et al., 2020), synthetic natural language expression (Clark et al., 2020; Yanaka et al., 2021; Fu and Frank, 2024), symbolic-refined LLMs (Olausson et al., 2023; Quan et al., 2024), etc.

Recent studies have shown that neural networks can encode sparse neural-symbolic concepts without explicit symbolic injection, across areas like image embedding (Ren et al., 2022; Deng et al., 2021; Li and Zhang, 2023), word embedding (Ethayarajh et al., 2018; Allen et al., 2019; Ri et al., 2023), contextual embedding (Gurnee et al., 2023; Nanda et al., 2023; Park et al., 2023; Li et al., 2024), and LLM interpretation (Templeton et al., 2024). However, whether transformer-based NLI models can exhibit symbolic behavior remains underexplored, and this is the focus of our work.

Latent variable models in NLP. Latent variable models, such as VAE (Kingma and Welling, 2013), have shown the capability of symbolic representation, control, and interpretation over the latent sentence space where the dimensions are orthogonality and features are casually separated, which are widely deployed in the NLP domain, such as disentangled representation learning (Zhang et al., 2024a), style-transfer (Liu et al., 2023; Gu et al., 2023), etc. Comparatively, latent variable models are less utilised within NLI settings where fine-grained symbolic control is required (Zhang et al., 2024b). Thus, we promote this general concept by eliciting the latent variable model potential inherent in contemporary gradient-based NLI models.

Latent symbolic control via label. One approach to achieving symbolic control in latent variable models involves conditioning on well-defined domain features, using architectures like CVAEs (Carvalho et al., 2023) and diffusion models (Dhariwal and Nichol, 2021; Ho and Salimans, 2022). These methods allow for localized generation by injecting labels into the latent space, a technique widely used in the image domain. Similarly, in the mathematical inference domain (Meadows et al., 2023), label information is needed to guide the gen-

[1] https://anonymous.4open.science/r/Inference_type-5E07/

eration of conclusions from premise expressions. Inspired by those studies, we propose to deliver symbolic NLI control through label-based inference behaviour. In the next section, we are going to introduce the labels (inference types).

3 Defining Lexical Inference Types

Valentino et al. (2021) has demonstrated that step-wise explanation-based NLI cannot be directly framed as pure logical reasoning. Explanatory chains, while looking plausible at first inspection, commonly have subtler incompleteness and consistency problems from a logical point of view. Meanwhile, explanatory chains correspond to definable inference patterns and symbolic operations can be localised over the sentence structure. Motivated by this middle ground between logical representations and lexico-semantic inference patterns, we introduce granular inference types based on explanatory sentences, using AMR to define the symbolic operations involved in step-wise inference, linking transformations from premises to conclusions [2]. Table 1 describes the AMR-grounded inference types and examples from the EntailmenBank corpus. Next, we define each lexico-semantic inference type and the corresponding symbolic forms.

The substitution category refers to obtaining a conclusion by replacing a predicate/argument term from one premise with a predicate/argument term from the other premise. Possible variations of this category include (1) *argument (ARG) substitution*, (2) *predicate (PRED) substitution*, and (3) *frame (PRED+ARG) substitution*. In this category, one premise is used to connect two terms which are usually connected by *is a kind of, is a source of*, etc. Conceptualising the AMR representation as a graph, this can be symbolically represented as a subgraph substitution operation over the premise graphs, as illustrated in Figure 2. The *PRED substitution* category works in a similar manner, but replacing a predicate term. The two predicates are usually linked by the following patterns: "v_1 *is a kind of* v_2", "*to* v_1 *something means to* v_2 *something*", etc. The *frame (PRED+ARG) substitution* category combines both previous categories by replacing a frame (predicate subgraph) of one of the premises with one from the other premise.

[2]Please note that AMR is not used as a representation mechanism in the proposed architecture, but only to precisely ground these symbolic operations within a well-defined semantic representation structure.

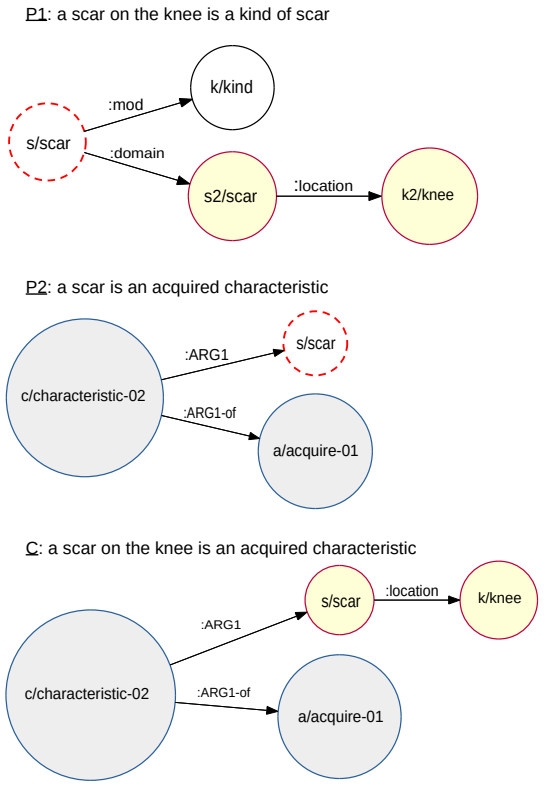


Figure 2: AMR argument substitution: the inference behaviour is defined as subgraph substitution.

The *further specification* or *conjunction* category allows for obtaining a conclusion by joining both premises. It includes (4) *ARG insertion* and (5) *frame conjunction*. In the case of *ARG insertion*, the conclusion is obtained by connecting an argument from one of the premises to a frame of the other. As for *frame conjunction/disjunction*, the conclusion is obtained by joining the premises graphs through a conjunction/disjunction node (*and*) or (*or*). The *inference from rule* category from (Dalvi et al., 2021) comprises a specific case of insertion or substitution categorised as (6) *conditional frame insertion/substitution*, where a frame is inserted or replaced as an argument of a premise following a conditional path in the other premise, as illustrated in Figure 5.

The inference type *infer class from properties* has been re-categorised as (7) *ARG or PRED generalisation*, where a new *:domain* relation frame is created if both premise graphs differ by a single predicate/argument term. (8) *Property inheritance*, on the other hand, is a special case of *ARG substitution*, where one of the premises describes a *is made of* relationship between the entity in the other premise and its replacement. Finally, the *other inference types* category includes (9) *example*, (10)

Original type	AMR type	Prop.	Example entailment relation
Substitution	ARG substitution (ARG-SUB)	19%	P1: a scar on the knee is a kind of scar P2: a scar is an acquired characteristic C: a scar on the knee is an acquired characteristic
	PRED substitution (PRED-SUB)	5%	P1: food contains nutrients and energy for living things P2: to contain something can mean to store something C: food stores nutrients and energy for living things
	Frame substitution (FRAME-SUB)	20%	P1: the formation of diamonds requires intense pressure P2: the pressure is intense deep below earth 's crust C: the formation of diamonds occurs deep below the crust of the earth
Inference from Rule	Conditional frame insertion/substitution (COND-FRAME)	12%	P1: if something is renewable then that something is not a fossil P2: fuel wood is a renewable resource C: wood is not a fossil fuel
Further Specification or Conjunction	ARG insertion (ARG-INS)	18%	P1: solar energy comes from the sun P2: solar energy is a kind of energy P3: solar energy is a kind of energy that comes from the sun
	Frame conjunction (FRAME-CONJ)	6%	P1: photosynthesis stores energy P2: respiration releases energy C: photosynthesis stores energy and respiration releases energy
Infer Class from Properties	ARG/PRED generalisation (ARG/PRED-GEN)	1%	P1: rock is a hard material P2: granite is a hard material C: granite is a kind of rock
Property Inheritance	ARG substitution (Property Inheritance) (ARG-SUB-PROP)	0.4%	P1: blacktop is made of asphalt concrete P2: asphalt has a smooth surface C: a blacktop has a smooth surface
Unknown	Example (EXAMPLE)	0.9%	a shelter can be used for living in by raccoons some raccoons live in hollow logs an example of a shelter is a raccon living in a hollow log
	If ... then ... (IFT)	0.8%	an optical telescope requires visible light for human to use clouds / dusts block visible light if there is clouds or dusts, then the optical telescope cannot be used
	Others (UNK)	16%	spiral is a kind of shape galaxies can be classified by shape spiral galaxy is a type of galaxy

Table 1: Example of AMR-based inference types. Their abbreviations are described in the parentheses, which is used in the paper. The size of the EntailmentBank corpus is 5134 in our task.

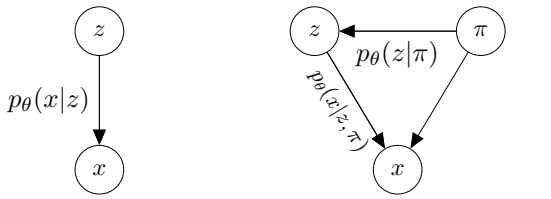
if-then, and (11) *others*. Those inference types are defined according to the key lexical characteristic of the conclusion, as systematic AMR transformations which could be applied without rephrasing the underlying explanatory sentences could not be determined. More details about this category and annotation procedure are provided in Appendix A.

4 Latent Variable NLI Framework

In this section, we first frame the NLI model as a latent variable model and align the notion of inference types as a conditional mechanism to support the learning and control of a quasi-symbolic latent space in section 4.1. Then, we instantiate the mechanism within a supporting neural architecture for learning latent sentence space in section 4.2.

4.1 Latent variable NLI

Recent work revealed that transformer-based language models can linearly encode abstract-level semantic concepts (latent variables, denoted by z) (Park et al., 2023; Li et al., 2024; Wang et al., 2024; Jiang et al., 2024). Following prior studies, we frame gradient-based neural NLI models as conditional latent variable models that can realise a quasi-symbolic inference dynamics. Assuming premises and conclusions share the same latent space where the explanatory entailment relation is computed in a probabilistic fashion, this allows for the framing of the entailment determination as the problem of learning a set of conditional probabilities among the latent variables. Figure 3 depicts an abstraction of the computational graph of the latent NLI/explanatory entailment framework.



(a) Latent NLI model (b) Conditional NLI model

Figure 3: Latent variable NLI framework.

Latent variables and relations. We first propose a set of latent variables based on prior studies of Zhang et al. (2024a), which revealed that explanatory sentence semantics can be decomposed into *semantic role - word content* sets (denoted by role-content) according to Argument Structure Theory (AST) (Jackendoff, 1992). E.g., the sentence, 'animals require oxygen for survival', can be represented as: $\underbrace{animals}_{ARG0} \oplus \underbrace{require}_{PRED} \oplus \underbrace{oxygen}_{ARG1} \oplus \underbrace{for\ survival}_{ARGM-PRP}$ where $\oplus$ represents the composition operation under a compositional-distributional model (Clark et al., 2008). Each role-content set, such as *ARG0-animals*, is encoded as a convex cone in the latent space. Therefore, we consider each role-content as a latent variable. The latent representation of observed sentence x can be formalised as a set of latent variables: $x \leftrightarrow z^{(x)} = \{(c_1, r_1), \ldots, (c_i, r_i), \ldots\}$ where $\leftrightarrow$ represent the deterministic mapping between x and $z^{(x)}$ through the embedding layer, $c_i \in C$ and $r_i \in R$ represent the word content and semantic role at position i, C and R are the vocabularies of word content and semantic role category, predefined based on training corpus.

Since an AMR representation is a particular instance of an AST representation, which represents the relation between latent variables, by defining and manipulating the inference patterns over the AMR representation in the context of inference types, we can provide quasi-symbolic interpretation and control to the latent NLI model. In the next section, the targeted NLI task supported by the AMR-grounded inference types is formalised under a Bayesian inference framework.

Latent Bayesian inference. In the context of step-wise, AMR-grounded explanations, given a premise-conclusion explanatory sentence pair $< x_{p_0}, x_{p_1}, x_c >$, an inference type $\pi \in \Pi$ can be associated, if exists a transformation $amr(x_{p_0}), amr(x_{p_1}) \rightarrow amr(x_c)$ defined over the set of transformations Π. The NLI process can be described as a Bayesian inference: $P(x_c|x_{p_0}, x_{p_1}) = P(x_c|z^{(x_c)})P(z^{(x_c)}|x_{p_0}, x_{p_1})$ where $P(z^{(x_c)}|x_{p_0}, x_{p_1})$ approximates the posterior inference via the encoder. Specifically, it first transform x_{p_0}, x_{p_1} into latent representations $z^{x_{p_0}}, z^{x_{p_1}}$. Subsequently, inference behaviour π is performed over the set of latent variables (e.g., substitution over latent variables set). The latent variables $z^{(x_c)}$ are retained for generation conclusion via decoder $P(x_c|z)$. To validate this inference process, we propose Proposition 1.

Proposition 1: *The inference behaviour is materialised during the posterior inference stage and can be controlled by the injection of the associated inference type labels,* Π*, into the posterior. That is the conditional inference process:*

$$P(x_c|x_{p_0}, x_{p_1}, \pi) = P(x_c|z^{(x_c)})P(z^{(x_c)}|x_{p_0}, x_{p_1}, \pi)$$

The inference type can be injected into the model at different points (e.g. at the encoder or decoder) and can be manipulated over different inference types to validate Proposition 1, as evaluated in Section 5.2. Finally, optimising the language modelling task approximates the latent variable space Z. This can be formalised as: $P(x_c) = \prod_{i=1}^{N} P(c_i|c_{i-1}, \ldots, c_1, Z)$ where c_i represent the i-th token. The latent variable model can be used to deliver a symbolic-level control over the "sentence" space, i.e., syntactically independent spans of text (Li et al., 2022). Therefore, in the next section, we cover the sentence level representation $P(z|x_{p_0}, x_{p_1})$ and different architectural choices to encode the associated inference type controls.

4.2 Latent variable NLI architecture

In this section, we describe the methodological framework behind the construction of the latent sentence space within T5, shown in Figure 8. We first investigate the sentence representation setup (i.e, $P(z|x_{p_0}, x_{p_1})$), then, move to latent space injection (i.e., $P(x_c|z)$).

Latent sentence space: $P(z|x_1, x_2)$. While designing the sentence bottleneck, We compare the four most frequently used mechanisms to transform token embeddings into sentence embeddings:

(1) Mean pooling: calculating the mean of each dimension on all token embeddings and feeding the resulting vector into a multi-layer perceptron to obtain the sentence embedding. (2) multi-layer perceptron (MLP): applying an MLP to reduce the dimensionality of token embeddings, and the resulting embeddings are concatenated to form a single sentence embedding: $z = \text{concat}\Big[\text{MLP}_1(x_1); ...; \text{MLP}_T(x_T)\Big]$ where $\text{MLP}_i(x_i)$ represents the i-th neural network for input representation of token x_i, z is the latent sentence representation, and T is the maximum token length for a sentence. (3) multi-head attention:

feeding each token embedding into the multi-head attention and considering the first output embedding as the sentence embedding (Montero et al., 2021): $z = \text{MultiHead}\left(XW^q, XW^k, XW^v\right)[0]$ where $X = [x_1, ..., x_T]$ and W^q, W^k, and W^v are the weights for learning q, k, v embeddings in self-attention, respectively. (4) Sentence T5: re-loading the pre-trained sentence T5 (S-T5, Ni et al. (2021)).

Conditional generation: $P(x_c|z)$. We consider four strategies to inject sentence embeddings into the decoder. (1) Cross-attention input embedding (CA Input): reconstructing the token embeddings from a sentence representation and directly feeding them into the cross-attention layers of the decoder: $\hat{Y} = \text{MultiHead}\left(YW^q, \text{MLP}(z)W^k, \text{MLP}(z)W^v\right)$ where $\hat{Y}$ is the reconstruction of decoder input sequence $Y = [y_1, ..., y_K]$. (2) Cross-attention KV embedding (CA KV): instead of reconstructing the token embeddings, it consists of directly learning the Key and Value (Hu et al., 2022; Li et al., 2020), which is formalised as $\hat{Y} = \text{MultiHead}\Big(YW^q, \text{MLP}_k(z), \text{MLP}_v(z)\Big)$, where MLP_k and MLP_v are neural layers for learning k v embeddings. (3) Non-cross-attention input connection (NCA Input): reconstructing the token embeddings and element-wisely adding them with the input embeddings of the decoder (Fang et al., 2021). (4) Non-cross-attention output connection (NCA Output): adding the reconstructed token embeddings to the output embedding of the decoder.

5 Empirical Analysis

5.1 Analysing the bottleneck baselines

In this experiment, we analyze the performance of the sentence bottleneck choices on the explanatory NLI task (i.e., syllogistic-deductive NLI: generating one conclusion from two premises) and the explanatory inference retrieval task, using T5 as a reference model.

Syllogistic-deductive NLI. Firstly, we evaluate different encoding-decoding configurations on the performance of sentence bottleneck[3]. As illustrated in Table 2 (top), we can observe that the best configuration is that the encoder is sentence T5 (S-T5), and the decoder is connected by reconstructing token embeddings as the input of cross-attention.

[3]We use the *base* version as the baseline for T5 and sentence bottleneck T5.

Next, we quantitatively evaluate the NLI performance of different baselines on the Entailment corpus. We specifically choose the VAE baselines, including the sota Transformer VAE model: Optimus (Li et al., 2020) and Della (Hu et al., 2022) with two different sentence dimensions (32 and 768), and five LSTM language autoencoders with 768 latent dimensions: denoising AE (Vincent et al. (2008), DAE), β-VAE (Higgins et al., 2016), adversarial AE (Makhzani et al. (2015), AAE), label adversarial AE (Rubenstein et al. (2018), LAAE), and denoising adversarial autoencoder (Shen et al. (2020), DAAE). In Table 2 (bottom), we can observe that our T5bottleneck can outperform all baselines on BLEU (Papineni et al., 2002), BLEURT (Sellam et al., 2020), and cosine similarity from pre-trained sentence T5 (Ni et al., 2021).

Train: architecture					
Decoder Connection		CA Input	CA KV	NCA Input	NCA Output
Sentence Embedding	Pooling	1.41	1.44	1.86	2.42
	MLP	1.71	1.94	2.09	2.62
	MHA	1.51	2.24	2.31	3.03
	S-T5	**1.24**	1.42	1.81	2.22
Test: EntailmentBank					
Metrics	BLEU	Cosine	BLEURT	Loss ↓	PPL ↓
Optimus(32)	0.07	0.74	-1.20	1.13	2.31
Optimus(768)	0.08	0.74	-1.21	**0.82**	**2.27**
DELLA(32)	0.08	0.85	-1.23	1.69	5.41
DELLA(768)	0.09	0.87	-1.09	1.54	4.66
DAE(768)	0.15	0.89	-0.95	1.33	3.78
AAE(768)	0.11	0.88	-0.95	1.35	3.85
LAAE(768)	0.09	0.74	-1.12	1.38	3.97
DAAE(768)	0.07	0.74	-1.20	1.43	4.17
β-VAE(768)	0.07	0.74	-1.20	1.43	4.17
T5bottleneck	**0.35**	**0.91**	**-0.20**	1.24	3.45

Table 2: Top: comparison of best setup on test loss via cross-entropy (CA: cross-attention, NCA: non-cross-attention), bottom: comparison of different baselines on EntailmentBank testset.

depth	t=1	t=2	t=3	t=4
DAE(768)	30.27	31.74	30.65	30.74
AAE(768)	29.13	30.47	29.33	29.14
LAAE(768)	19.13	20.86	18.32	18.01
DAAE(768)	13.16	15.42	14.30	13.97
β-VAE(768)	10.03	10.07	10.05	10.05
Optimus(768)	28.21	29.35	28.35	28.27
T5 bottleneck(768)	**34.47**	**35.28**	**34.50**	**34.47**

Table 3: Explanatory inference retrieval task where t represents the depth of entailment tree.

Explanatory inference retrieval. Furthermore, we evaluate the sentence embedding using as an associated explanation retrieval task (explanation-

regeneration - i.e. retrieving the associated explanatory facts relevant to a claim) (Valentino et al., 2022a). Given a question-and-answer pair, it reconstructs the entailment tree by searching the explanations from a fact bank (i.e., WorldTree (Jansen et al., 2018)) in an iterative fashion using a dense sentence encoder. In this framework, we can replace the dense sentence encoder with the proposed T5 bottleneck baseline to evaluate its sentence embeddings. We compare the T5 bottleneck with sentence VAEs: Optimus and five LSTM VAEs, and evaluate them via mean average precision (MAP). As illustrated in Table 3, the T5 bottleneck outperforms all baselines, indicating that it can deliver a better representation of explanatory sentences and entailment relations in a retrieval setting.

5.2 Analysing the impact of inference types

Next, we evaluate the performance of the model with integrated inference types by comparing the following baselines: 1. seq2seq: T5 family and Bart (Lewis et al., 2019), 2. casualLM: GPT family (Radford et al., 2019) and LlaMA3 and 3. seq2seq with sentence bottleneck: T5 bottleneck and Optimus. We focus on addressing two questions:

1. Can the inference types improve model training and inference? If so, models trained (or prompted) with inference types should exhibit higher scores (such as BLEU and BLEURT) on the test set.

2. Can inference types be used for a prescriptive inference control? If the association between inference types and the corresponding symbolic transformations are learned, manipulating the inference type could achieve a controlled generation.

We consider three mechanisms to conditionally inject the inference types into the latent space, which are described below, where *p1*, *p2*, and *con* are the premises and conclusion, respectively, and </s> is a special token for sentence separation.

i. The inference type as the prefix for the premises at the Encoder: *the inference type is [type] </s> p1 </s> p2*

ii. The inference type as the prefix for the conclusion in the Decoder: *</s> the inference type is [type]. con*

iii. The inference type at the end of the conclusion in the Decoder: *</s> con. the inference type is [type]*

Fine-tuning. We first quantitatively evaluate model performance based on five metrics: test loss (cross-entropy), perplexity (PPL), BLEURT, BLEU, and cosine similarity against sentenceT5. As illustrated in Table 4, all baselines with inference types always have lower test losses and PPLs, which means the inference type can help the model training. Furthermore, across all baseline models, incorporating inference types into the encoder consistently results in improved performance as measured by BLEU, Cosine, and BLEURT metrics. This finding suggests that the conditionalisation on inference types can support the inference representation, and the inference process has been performed inside the encoder (*Proposition1*).

Baseline	INJ	BLEU	Cosine	BLEURT	Loss ↓	PPL ↓
seq2seqLM: encoder-decoder architecture						
T5 original (small)	DE	0.55	0.96	0.30	0.53	1.44
	DP	0.59	0.96	0.34	0.58	1.57
	EP	**0.65**	**0.97**	**0.45**	**0.52**	**1.41**
	NO	0.54	0.96	0.22	0.69	2.22
T5 original (base)	DE	0.46	0.96	0.23	0.49	1.33
	DP	0.53	0.96	0.25	0.51	1.38
	EP	**0.61**	**0.97**	**0.39**	**0.45**	**1.22**
	NO	0.57	0.96	0.33	0.61	1.65
Bart (base)	DE	0.44	0.94	0.03	0.55	1.49
	DP	0.38	0.93	-0.42	**0.48**	**1.30**
	EP	**0.57**	**0.96**	**0.23**	0.58	1.57
	NO	0.54	0.96	0.17	0.63	1.71
T5 original (large)	DE	0.60	0.97	0.46	**0.40**	1.49
	DP	0.64	0.97	0.44	0.46	1.58
	EP	**0.67**	**0.97**	**0.50**	0.59	1.80
	NO	0.57	0.96	0.31	0.61	1.84
Flan-T5 (large)	DE	0.01	0.73	-1.34	6.91	10.2
	DP	0.01	0.73	-1.34	7.00	15.4
	EP	**0.21**	**0.87**	**-1.04**	**1.30**	**3.66**
	NO	0.20	0.87	-1.14	1.34	3.81
T5 original (3b)	DE	0.60	0.96	0.44	0.68	1.97
	DP	0.66	0.96	0.49	0.65	1.91
	EP	**0.70**	**0.97**	**0.57**	**0.51**	**1.66**
	NO	0.68	0.97	0.55	0.63	1.87
CausalLM: decoder only architecture						
GPT2 (large)	DE	0.02	0.87	-1.15	0.73	2.07
	DP	**0.08**	**0.90**	**-0.91**	**0.73**	**2.07**
	NO	0.07	0.90	-0.93	0.76	2.06
GPT2 (xl)	DE	0.20	0.88	-1.10	0.63	1.87
	DP	**0.28**	**0.91**	**-0.90**	**0.60**	**1.82**
	NO	0.27	0.90	-0.97	0.68	1.97
seq2seqLM with sentence bottleneck						
T5 bottleneck (base)	DE	0.35	0.91	-0.15	**0.84**	**2.31**
	DP	0.39	0.91	-0.13	0.86	2.36
	EP	**0.42**	**0.92**	**-0.07**	1.23	3.42
	NO	0.35	0.91	-0.20	1.24	3.45
Optimus (BERT-GPT2)	DE	**0.26**	**0.80**	**-1.11**	0.87	2.38
	DP	0.25	0.79	-1.14	**0.85**	**2.33**
	EP	0.09	0.74	-1.17	1.11	3.03
	NO	0.07	0.74	-1.20	1.13	3.09

Table 4: Quantitative evaluation of test cases, where best results are highlighted in **bold**. Specification for abbreviation. INJ: ways for injecting the information of inference types into the model, it includes DE: decoder end, DP: decoder prefix, EP: encoder prefix, NO: no inference type. PPL is perplexity, Loss is cross entropy.

In-context learning. Next, we quantitatively evaluate the inference types within in-context learning (ICL) in contemporary large language models

Baseline	INJ	Num	BLEU	Cosine	BLEURT
Seq2seqLLM: encoder-decoder architecture					
CoT-T5 (11b) (Kim et al., 2023)	Yes	10	0.51	0.97	0.39
	Yes	5	0.51	0.97	0.39
	Yes	0	0.50	0.97	0.36
	NO	0	**0.46**	**0.96**	**0.31**
Flan-T5 (xl)	Yes	10	0.49	0.96	0.40
	Yes	5	0.48	0.96	0.39
	Yes	0	0.52	0.96	0.39
	NO	0	**0.44**	**0.95**	**0.24**
Flan-T5 (xxl)	Yes	10	0.51	0.97	0.41
	Yes	5	0.53	0.97	0.43
	Yes	0	0.50	0.96	0.37
	NO	0	**0.48**	**0.96**	**0.36**
CausalLLM: decoder only architecture					
GPT-3.5-turbo-0125	Yes	10	0.52	0.96	0.40
	Yes	5	0.48	0.96	0.35
	Yes	0	0.46	0.96	**0.31**
	NO	0	**0.42**	0.96	0.33
GPT-4-0613	Yes	10	0.53	0.97	0.50
	Yes	5	0.52	0.97	0.47
	Yes	0	0.52	0.97	0.50
	NO	0	**0.47**	**0.96**	**0.40**
llama3-8b-8192	Yes	10	0.48	0.96	0.33
	Yes	5	0.45	0.96	0.32
	Yes	0	0.37	0.95	0.22
	NO	0	**0.34**	**0.95**	**0.19**
llama3-70b-8192	Yes	10	0.54	0.97	0.54
	Yes	5	0.52	0.97	0.52
	Yes	0	0.51	0.97	0.47
	NO	0	**0.44**	**0.96**	**0.40**

Table 5: ICL evaluation of test cases, where worst results are highlighted in **bold**. The prompt is *"performing natural language inference [where the inference type is type, description],* $[p1; p2; c]_{\times Num}$*. p1, p2, what is the conclusion?"*. The *description* is based on the definition of inference types in Section 3.

(LLMs). As illustrated in Table 5, prompting with inference types can improve the performance of ICL in both seq2seq and causal LLMs. Besides, within the context of causal LLMs, an increase in few shot examples[4], improves the performance.

Controllability evaluation. Moreover, we qualitatively[5] evaluate the controllability on the generation of conclusions by systematically intervening on the inference type prior to the encoder. As illustrated in Table 6, we can observe that the associated linguistic properties of the conclusion can be controlled consistently with the inference type modifications, which indicates that the representation mechanisms can improve inference control with regard to symbolic/lexico-semantic properties. For example, when the type is ARG substitution (ARG-SUB), the model can generate *the blacktop is made of a smooth surface* by replacing the argument *asphalt concrete* with *smooth surface*. The conclusions are changed to *asphalt and blacktop have the same surface* when the inference type is the conjunction (FRAME-CONJ). More examples are provided in Table 11.

P1: blacktop is made of asphalt concrete
P2: asphalt has a smooth surface

ARG-SUB: the blacktop is made of smooth surface
ARG-SUB-PROP: blacktop has a smooth surface
ARG/PRED-GEN: a blacktop is a kind of asphalt
ARG-INS: asphalt concrete blacktop has a smooth surface
FRAME-CON: asphalt and blacktop have the same surface
IFT: if the asphalt has a smooth surface then the blacktop will have a smooth surface

Table 6: Controllable generation over original T5 (base) (ARG-SUB: argument substitution, ARG/PRED-GEN: argument/predicate generalisation. ARG-SUB-PROP: property inheritance. ARG-INS: argument insertion, FRAME-CON: frame conjunction, IFT: if...then...). The example of the T5 bottleneck is provided in Table 9.

6 Conclusion and Future Work

In this work, we focus on the syllogistic-deductive NLI inference task to explore symbolic-level inference control on language models. We first define a set of universal inference types for capturing inference rules behind explanatory sentences, grounded on AMR graphs, as a mechanism to precisely define symbolic inference operations.

To integrate symbolic-level annotations into the NLI model, we formalise the neural NLI model as a latent variable model, enabling a bridge between symbolic and distributional representations. Based on this framework, our symbolic inference control is modelled as learning a conditional latent variable model. Besides, as a complementary, we propose T5 Bottleneck architecture to enhance the research on sentence representations.

Experimental results revealed that the proposed approach can improve model training, inference, in-context learning, retrieval, and symbolic control. Based on our findings, in the future, we will explore the dynamics of symbolic explanatory inference patterns inside language models to enhance their mechanism interpretability in the NLI task (Deng et al., 2021; Li and Zhang, 2023).

[4]We randomly sample the examples with the same inference type as the current test example from the training set. We perform ten times and calculate the average for each metric.

[5]The quantitative evaluation cannot be automatically performed since some premises can only derive one conclusion.

7 Limitations

While the work focuses on the controllability of explanatory inference, complementary methods need to be employed to deliver more strict safety guarantees. In short, control and safety should not be confounded and should be delivered by independent mechanisms.

Quantitative evaluation of controllability depends on the elaboration of a new ground-truth dataset, mapping the valid conclusions to a set of premises and the corresponding inference types. This is a direction to be explored in future work.

References

Lasha Abzianidze. 2023. Formal proofs as structured explanations: Proposing several tasks on explainable natural language inference. *arXiv preprint arXiv:2311.08637*.

Carl Allen, Ivana Balazevic, and Timothy Hospedales. 2019. What the vec? towards probabilistically grounded embeddings. *Advances in neural information processing systems*, 32.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Danilo S. Carvalho, Yingji Zhang, Giangiacomo Mercatali, and Andre Freitas. 2023. Learning disentangled representations for natural language definitions. In *Findings of the European chapter of Association for Computational Linguistics (Findings of EACL)*. Association for Computational Linguistics.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, pages 133–140. Oxford.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. *arXiv preprint arXiv:2104.08661*.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of EACL*.

Huiqi Deng, Qihan Ren, Hao Zhang, and Quanshi Zhang. 2021. Discovering and explaining the representation bottleneck of dnns. *arXiv preprint arXiv:2111.06236*.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2018. Towards understanding linear word analogies. *arXiv preprint arXiv:1810.04882*.

Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Transformer-based conditional variational autoencoder for controllable story generation. *arXiv preprint arXiv:2101.00828*.

Xiyan Fu and Anette Frank. 2024. Exploring continual learning of compositional generalization in nli. *arXiv preprint arXiv:2403.04400*.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, Weihong Zhong, and Bing Qin. 2023. Controllable text generation via probability density estimation in the latent space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12590–12616, Toronto, Canada. Association for Computational Linguistics.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *arXiv preprint arXiv:2305.01610*.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Jinyi Hu, Xiaoyuan Yi, Wenhao Li, Maosong Sun, and Xing Xie. 2022. Fuse it more deeply! a variational transformer with layer-wise latent variable inference for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 697–716, Seattle, United States. Association for Computational Linguistics.

Ray S Jackendoff. 1992. *Semantic structures*, volume 18. MIT press.

Peter Jansen, Rebecca Sharp, Mihai Surdeanu, and Peter Clark. 2017. Framing qa as building and ranking intersentence answer justifications. *Computational Linguistics*, 43(2):407–449.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In

Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. 2024. On the origins of linear representations in large language models. *arXiv preprint arXiv:2403.03867*.

Aikaterini-Lida Kalouli, Richard Crouch, and Valeria de Paiva. 2020. Hy-NLI: a hybrid system for natural language inference. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5235–5249, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. *arXiv preprint arXiv:1704.05572*.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.14045*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Mingjie Li and Quanshi Zhang. 2023. Does a neural network really encode symbolic concepts? In *International Conference on Machine Learning*, pages 20452–20469. PMLR.

Zhuang Li, Lizhen Qu, Qiongkai Xu, Tongtong Wu, Tianyang Zhan, and Gholamreza Haffari. 2022. Variational autoencoder with disentanglement priors for low-resource task-specific natural language generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10335–10356, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William W Cohen. 2020. Differentiable open-ended commonsense reasoning. *arXiv preprint arXiv:2010.14439*.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. 2023. Composable text controls in latent space with ODEs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16543–16570, Singapore. Association for Computational Linguistics.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Jordan Meadows, Marco Valentino, Damien Teney, and Andre Freitas. 2023. A symbolic framework for systematic evaluation of mathematical reasoning with transformers. *arXiv preprint arXiv:2305.12563*.

Ivan Montero, Nikolaos Pappas, and Noah A Smith. 2021. Sentence bottleneck autoencoders from transformer language models. *arXiv preprint arXiv:2109.00055*.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5153–5176, Singapore. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Xin Quan, Marco Valentino, Louise A Dennis, and André Freitas. 2024. Verification and refinement of natural language explanations through llm-symbolic theorem proving. *arXiv preprint arXiv:2405.01379*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Jie Ren, Mingjie Li, Qirui Chen, Huiqi Deng, and Quanshi Zhang. 2022. Towards axiomatic, hierarchical, and symbolic explanation for deep models.

Narutatsu Ri, Fei-Tzin Lee, and Nakul Verma. 2023. Contrastive loss is all you need to recover analogies as parallel lines. *arXiv preprint arXiv:2306.08221*.

Paul K Rubenstein, Bernhard Schoelkopf, and Ilya Tolstikhin. 2018. On the latent space of wasserstein auto-encoders. *arXiv preprint arXiv:1802.03761*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi Jaakkola. 2020. Educating text autoencoders: Latent representation guidance via denoising. In *International Conference on Machine Learning*, pages 8719–8729. PMLR.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020. A survey on explainability in machine reading comprehension. *arXiv preprint arXiv:2010.00389*.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Explainable inference over grounding-abstract chains for science questions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1–12.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2024. A differentiable integer linear programming solver for explanation-based natural language inference. *arXiv preprint arXiv:2404.02625*.

Marco Valentino, Ian Pratt-Hartmann, and André Freitas. 2021. Do natural language explanations represent valid logical arguments? verifying entailment in explainable nli gold standards.

Marco Valentino, Mokanarangan Thayaparan, Deborah Ferreira, and André Freitas. 2022a. Hybrid autoregressive inference for scalable multi-hop explanation regeneration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11403–11411.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2020. Explainable natural language reasoning via conceptual unification. *arXiv preprint arXiv:2009.14539*.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2022b. Case-based abductive natural language inference. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1556–1568.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA. Association for Computing Machinery.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. SyGNS: A systematic generalization testbed based on natural language semantics. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 103–119, Online. Association for Computational Linguistics.

Yingji Zhang, Danilo Carvalho, and Andre Freitas. 2024a. Learning disentangled semantic spaces of explanations via invertible neural networks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2113–2134, Bangkok, Thailand. Association for Computational Linguistics.

Yingji Zhang, Danilo Carvalho, Marco Valentino, Ian Pratt-Hartmann, and Andre Freitas. 2024b. Improving semantic control in discrete latent spaces with transformer quantized variational autoencoders. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1434–1450, St. Julian's, Malta. Association for Computational Linguistics.

A Annotation details

Other inference types. the *other inference types* category includes (9) *example*, (10) *if-then*, and (11) *others*. Those inference types are defined according to the key lexical characteristic of the conclusion, as systematic AMR transformations which could be applied without rephrasing the underlying explanatory sentences could not be determined. For example, the *if-then* category refers to a conclusion that has an *if then* sentence structure in which both premises do not contain it. The *example* category is a conclusion with the signalling word *example*, which does not appear in either premise. As for other unknown categories, we do not further specify them, as they either require knowledge outside of the scope of the premises or do not have a consistent symbolic transformation expression. An additional subtype called *premise copy* was included for the cases where the conclusion has the same graph as one of the premises.

Annotation procedure. Annotation was performed manually for 5134 entailment triples (two premises, one conclusion) from the EntailmentBank (Dalvi et al., 2021), according to Algorithm 1. Graph subset relations and root matching are relaxed for non-argument (:ARG*, op*) edges, meaning relations such as *:manner* or *:time* can be ignored for this purpose. Two independent annotators with post-graduate level backgrounds in Computational Linguistics were used in this process, on a consensus-based annotation scheme where a first annotator defined the transformations and a second annotator verified and refined the annotation scheme, in two iterations. The annotation of the AMR graph is based on an off-the-shelf parser (Damonte et al., 2017). The descriptions for each inference type category are as follows:

ARG-SUB (Figure 2): the conclusion is obtained by replacing one argument with another argument.

PRED-SUB: the conclusion is obtained by replacing one verb with another verb.

FRAME-SUB: the conclusion is obtained by replacing a frame of one of the premises with one from the other premise.

COND-FRAM (Figure 5): the conclusion is obtained according to the conditional premise with keyword "if".

ARG-INS (Figure 4): the conclusion is obtained by connecting an argument from one of the premises to a frame of the other.

FRAME-CONJ: the conclusion is obtained by using connectives to connect two premises.

ARG/PRED-GEN (Figure 6): a new *:domain* relation frame is created in the conclusion if both premise graphs differ by a single predicate/argument term.

ARG-SUB-PROP (Figure 7): one of the premises describes a "*is made of*" relationship between the entity in the other premise and its replacement.

IFT: the conclusion should be a conditional sentence.

EXAMPLE: the conclusion should contain the keyword "example".

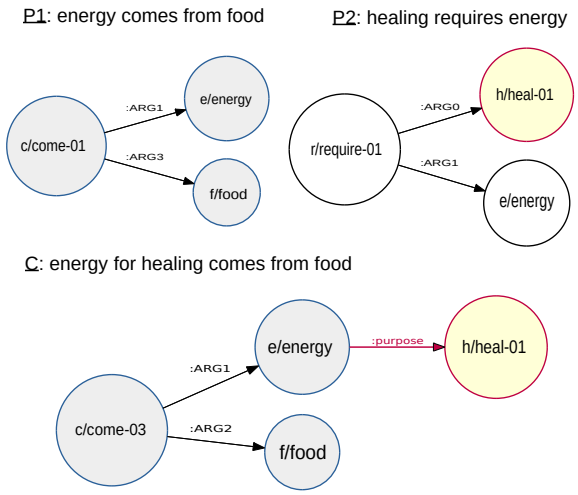


Figure 4: AMR argument insertion (ARG-INS).

B Implementation details

Architecture overview. Figure 8 shows the architecture of T5 bottleneck for learning latent sentence space. It includes two stages: sentence embedding and decoder connection. The sentence embedding aims to transform token embeddings into a sentence (single) embedding. Decoder connection aims to connect the encoder and decoder.

Implementation details. **1.** Size of Sentence Representation: in this work, we consider 1000 as the size of the sentence embedding. Usually, the performance of the model improves as the size increases. **2.** Multi-head Attention (MHA): in the experiment, MHA consists of 8 layers, each layer containing 12 heads. The dimensions of Query, Key, and Value are 64 in each head. The dimension of token embedding is 768. Training hyperparameters are: **3.** Max epoch: 40, learning rate: 5e-5.

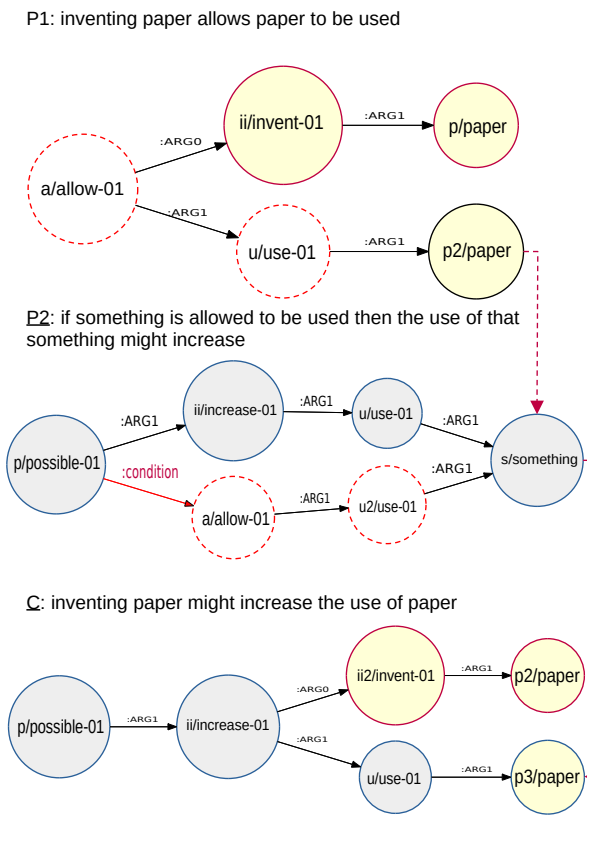


Figure 5: AMR conditional frame insertion (COND-FRAME).

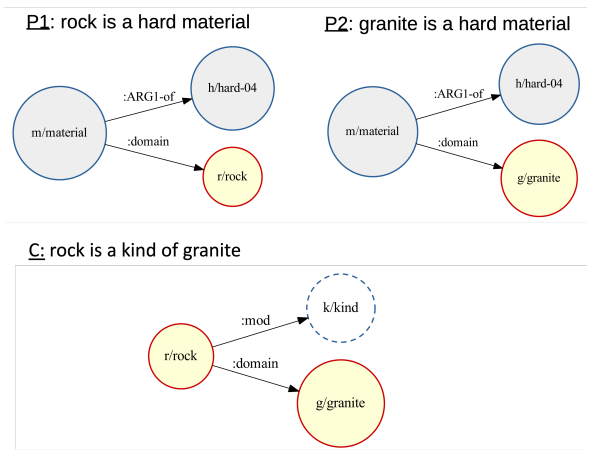


Figure 6: AMR argument generalisation (ARG-GEN).

During fine-tuning the sentence bottleneck T5, we first freeze the pre-trained parameters in the first epoch and fine-tune all parameters for the remaining epochs. **4.** The implementation of Optimus is based on their original code (Li et al., 2020) [6]. It is initialised with a pretrained checkpoint with β=0.

[6] https://github.com/ChunyuanLI/Optimus

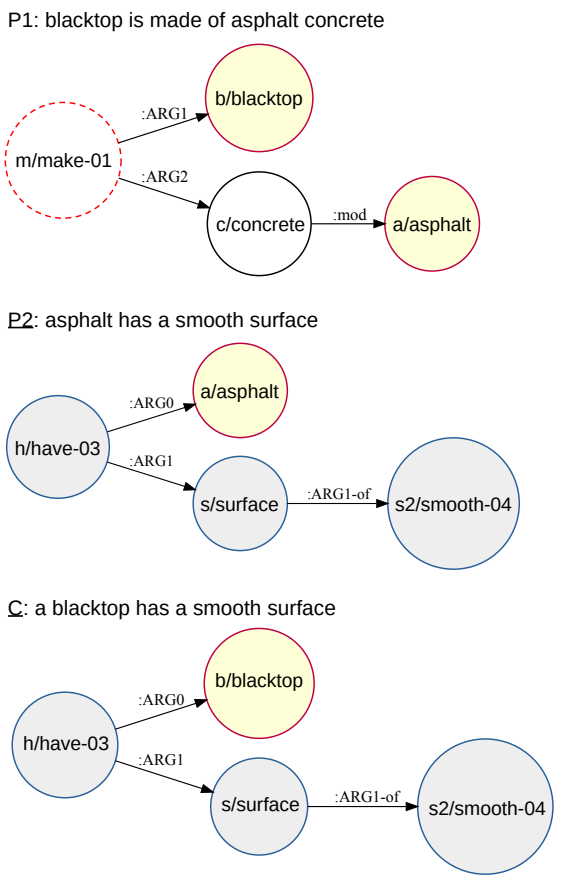


Figure 7: AMR argument substitution (property inheritance) (ARG-SUB-PROP).

Size	32	64	128	256	512	1000
Loss	1.59	1.42	1.36	1.32	1.29	1.26

Table 7: The loss with different sentence dim in sentence bottleneck T5.

Baselines. In the experiment, we implement five LSTM-based autoencoders, including denoising AE (Vincent et al. (2008), DAE), β-VAE (Higgins et al., 2016), adversarial AE (Makhzani et al. (2015), AAE), label adversarial AE (Rubenstein et al. (2018), LAAE), and denoising adversarial autoencoder (Shen et al. (2020), DAAE). Their implementation relies on the open-source codebase available at the URL [7]. As for transformer-based VAEs, we implement Optimus (Li et al., 2020) and Della (Hu et al., 2022)[8]. All baseline models undergo training and evaluation with the hyper-parameters provided by their respective sources. A latent dimension of 768 is specified to ensure a uniform and equitable comparative analysis.

[7] https://github.com/shentianxiao/text-autoencoders

[8] https://github.com/OpenVLG/DELLA

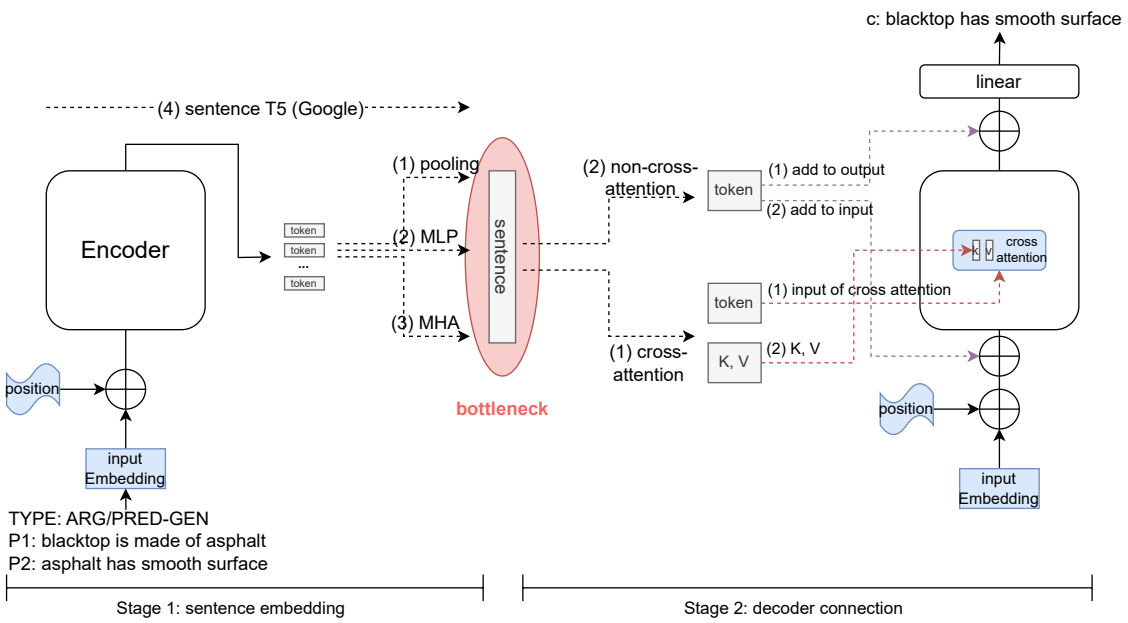


Figure 8: The architectural configuration of T5bottleneck.

C Complementary Results

Ablation studies. We remove the inference types from the dataset and evaluate the T5 model performance using the same metrics. In this case, we can compare the model performance trained with or without that inference-type data. From Table 8, we can observe that the baselines (T5 small and base) achieve higher BLEU and BLEURT scores without the data with further specification (FUR), Inference from rule (INF-RULE), and unknown (UNK) inference type, respectively. This result indicates that the T5 cannot generalize well over those inference types. Also, removing the UNK inference type from data can achieve lower loss and PPL, which indicates that it has a negative impact on model training.

Remove	T5	BLEU	BLEURT	Cosine	Loss ↓	PPL ↓
FRAME-SUB	small	0.50	0.19	0.95	0.95	2.58
	base	0.60	0.33	0.96	0.72	1.95
ARG-INS	small	**0.54**	**0.27**	0.95	0.82	2.22
	base	**0.63**	**0.46**	0.97	0.64	1.73
FRAME-CONJ	small	0.53	0.26	0.96	0.84	2.28
	base	0.60	0.35	0.96	0.65	1.76
COND-FRAME	small	**0.55**	**0.25**	0.96	0.88	2.39
	base	**0.59**	**0.36**	0.96	0.69	1.87
UNK	small	**0.55**	**0.23**	0.95	0.53	1.44
	base	**0.62**	**0.40**	0.96	0.58	1.57
No	small	0.54	0.22	0.96	0.69	2.22
No	base	0.57	0.33	0.96	0.61	1.65

Table 8: Ablation study over inference type (No: no inference types are removed).

More controllable inference examples. We provide more controlled examples based on both the Original T5 and T5 bottleneck in Table 9, 10, and 11. All examples reveal that the inference type can provide quasi-symbolic inference control to language models.

P1: a pumpkin contains seeds
P2: fruit contains seeds

Original T5:
ARG-INS: a fruit in a pumpkin contains seeds
FRAME-CONJ: a pumpkin and fruit both contains seeds
FRAME-SUB: fruit is a kind of pumpkin

T5 bottleneck:
ARG-INS: fruit is a part of pumpkin that contains seeds
FRAME-CONJ: a fruit contains seeds
FRAME-SUB: a pumpkin is a kind of plant

Table 9: Controlled generation. original T5(base) (top) and T5 bottleneck (bottom).

P1: eating something has a negative impact on that something
P2: some animals eat cacti
ARG-INS: some animals have a negative impact on cacti by eating cacti
PRED-SUB: some animals may have a negative impact on cacti
FRAME-SUB: eating cacti has a negative impact on that cacti

ARG-INS: some animals have a negative impact on cacti by eating cacti
PRED-SUB: animals have a negative impact on cacti
FRAME-SUB: eating cacti has a negative impact on that cacti

Table 10: Controlled generation. original T5(base) (top) and T5 bottleneck (bottom).

Premises	Inference Type	Original T5
P1: a pumpkin contains seeds	ARG-INS	a fruit in a pumpkin contains seeds
P2: fruit contains seeds	FRAME-CONJ	a pumpkin and fruit both contain seeds
	IFT	if a pumpkin contains fruit then the fruit may contain seeds
	EXAMPLE	fruit is an example of pumpkins being sown
	ARG/PRED-GEN	a pumpkin is a kind of fruit
	ARG-SUB	fruit can contain pumpkin seeds
	UNK	a pumpkin can contain seeds
	FRAME-SUB	fruit is a kind of pumpkin
P1: sunlight is a kind of solar energy	ARG-INS	solar energy is a kind of resource for plants that uses water
P2: water and sunlight are resources for plants	FRAME-CONJ	water and sunlight are resources for plants and are kinds of solar energy
	UNK	the resources for plants include water and solar energy
	ARG-SUB	water and solar energy are resources for plants
P1: to move something can mean to transfer something	ARG-SUB	flowing can mean to transfer energy
P2: flowing is a kind of movement for energy	INF	if something flows, then that energy will flow
	FRAME-CONJ	moving can transfer energy and mean flowing
	ARG-INS	flowing can be a kind of transfer of energy to another entity
	ARG/PRED-GEN	transferring energy with flowing can be seen as transferring energy
P1: if a pot is exposed to a stove then that pot may become hot	COND-FRAME	the pot will become hot
P2: the pot is exposed to a stove	EXAMPLE	an example of hot pot is a pot exposed to a stove
P1: eating something has a negative impact on that something	FRAME-SUB	eating cacti has a negative impact on that cacti
P2: some animals eat cacti	PRED-SUB	some animals may have a negative impact on cacti
	ARG-INS	some animals have a negative impact on cacti by eating cacti
	EXAMPLE	cooking cacti is an example of a negative impact on a cactus
	INF	if a cactus has a negative impact on an animal, that cactus could be devoured
P1: seeing requires light	ARG-SUB	reading requires light
P2: reading requires seeing	ARG-INS	light is a kind of requirement for reading
	INF	if light is moving then reading may be taken
	EXAMPLE	a light bulb will be used for reading
	UNK	light will help you read

Premises	Inference Type	T5 bottleneck
P1: a pumpkin contains seeds	ARG-INS	fruit is part of a pumpkin that contains seeds
P2: fruit contains seeds	FRAME-CONJ	a fruit contains seeds
	FRAME-SUB	a pumpkin is a kind of plant
P1: sunlight is a kind of solar energy	ARG-INS	water is a kind of resource that is used by plants for growth
P2: water and sunlight are resources for plants	FRAME-CONJ	plants and water are resources that require water and energy
	ARG-SUB	plants use water and sunlight to produce energy
P1: to move something can mean to transfer something	ARG-SUB	flowing can mean to transfer energy
P2: flowing is a kind of movement for energy	INF	if something flows, then that energy will flow
	FRAME-CONJ	moving can transfer energy and mean flowing
	ARG-INS	flowing can be a kind of transfer of something
	ARG/PRED-GEN	transferring energy with flowing can be seen as transferring energy
P1: if a pot is exposed to a stove then that pot may become hot	COND-FRAME	the pot may become hot
P2: the pot is exposed to a stove	ARG/PRED-GEN	the pot may be a source of heat
P1: eating something has a negative impact on that something	FRAME-SUB	eating cacti has a negative impact on that cacti
P2: some animals eat cacti	PRED-SUB	animals have a negative impact on cacti
	ARG-INS	some animals have a negative impact on cacti by eating cacti
P1: seeing requires light	ARG-SUB	reading requires light
P2: reading requires seeing	FRAME-CONJ	reading and feeling can both be used
	INF	if something is visible then that something will be seen

Table 11: controllable NLI via inference type (Top: original T5, bottom: T5 bottleneck).

Algorithm 1 Annotation procedure

1: Find premise P_x most similar to the conclusion C, $P_{\bar{x}}$ being the other premise.
2: $G_{x,\bar{x},C} \leftarrow$ AMR graph of $P_x, P_{\bar{x}}, C$, respectively.
3: # - - - - - - - - - - - - - - - - - - common ARG-SUB, PRED-SUB -
4: **if** $G_x = G_c$ or $G_{\bar{x}} = G_c$ **then**
5: $\quad type = PREM\text{-}COPY$ # Comment: no reasoning happen.
6: **else if** P_x and C differ by one word w **then** # Comment: common ARG(PRED)-SUB.
7: $\quad$ **if** w is a verb **then**
8: $\quad\quad type = PRED\text{-}SUB$
9: $\quad$ **else**
10: $\quad\quad type = ARG\text{-}SUB$
11: $\quad$ **end if**
12: **else**
13: # - - - - - - - - - - - - - - - - - COND-FRAME, FRAME-SUB, ARG-SUB-PROP - - - - - - - - - - - - - -
14: $\quad$ Get AMR graphs G_1, G_2, G_c for P_1, P_2 and C respectively. $P_x \rightarrow G_x$.
15: $\quad$ **if** $\exists$:ARG*$(x, a) \in C$ and $a \in P_{\bar{x}}$ **then**
16: $\quad\quad$ **if** $\exists$:condition($root(G_x)$, $root(G_{\bar{x}})$) **then**
17: $\quad\quad$ # Comment: see Figure 5, two root nodes are connected by :condition edge
18: $\quad\quad\quad type = COND\text{-}FRAME$
19: $\quad\quad$ **else if** $root(a)$ is a noun **then**
20: $\quad\quad\quad$ **if** $root(G_{\bar{x}})$ = "make-01" and $\exists$:ARG*($root(G_{\bar{x}})$, a) **then**
21: $\quad\quad$ # Comment: "make" as a trigger to classify ARG-SUB and property inheritance.
22: $\quad\quad\quad\quad type = ARG\text{-}SUB\text{-}PROP$
23: $\quad\quad\quad$ **else**
24: $\quad\quad\quad\quad type = ARG\text{-}SUB$ # ARG-SUB that was not caught by the simpler rule on line 10, due to Px differing from C by more than a single word
25: $\quad\quad\quad$ **end if**
26: $\quad\quad$ **else**
27: $\quad\quad\quad type = FRAME\text{-}SUB$
28: $\quad\quad$ **end if**
29: # - - - - - - - - - - - - - - - - - - Further-specification and Conjunction -
30: $\quad$ **else if** $G_x \subset G_c$ and $G_{\bar{x}} \subset G_C$ **then**
31: $\quad\quad type = FRAME\text{-}CONJ$
32: $\quad$ **else if** $\exists x, y$:domain($root(G_x)$, x) and :domain($root(G_{\bar{x}}$, y) and :op*("and", x) $\in G_c$ and :op*("and", y) $\in G_c$ **then** # Comment: using connectives 'and' to connect two premises
33: $\quad\quad type = FRAME\text{-}CONJ$
34: $\quad$ **else if** $G_x \subset G_c$ **then**
35: $\quad\quad d \leftarrow G_c - G_x$
36: $\quad\quad$ **if** $root(d)$ is a noun **then**
37: $\quad\quad\quad type = ARG\text{-}INS$ # Comment: inserting an argument.
38: $\quad\quad$ **else**
39: $\quad\quad\quad type = FRAME\text{-}INS$ # Comment: inserting a phase (also annotated as ARG-INS).
40: $\quad\quad$ **end if**
41: # - ARG/PRED-GEN and Others -
42: $\quad$ **else if** $\exists$:domain($root(G_c)$, y) and ($root(G_c) \in G_x$ and $y \in G_{\bar{x}}$) or ($root(G_c) \in G_{\bar{x}}$ and $y \in G_x$) **then**
43: $\quad\quad type = ARG/PRED\text{-}GEN$
44: $\quad$ **else**
45: $\quad\quad type = UNK$
46: $\quad$ **end if**
47: **end if**